Claims

1. A method for instrumenting virtual-machine-executable software, the software including one or more objects, each object being defined by a class, each class being associated with a class-hierarchy location, the method comprising the steps of:

(a) identifying at least one target class included within the software, the at least one target class being associated with a first class-hierarchy location and with a first class name;

(b) for each target class, adding instrumentation to the software according to at least one of the following steps:

(i) creating a new class, adding instrumentation to the new class, and assigning the new class to a class-hierarchy location adjacent to and above the first class-hierarchy location of the target class; and

(ii) creating a new class, adding instrumentation to the new class, and assigning the new class to a class-hierarchy location adjacent to and below the first class-hierarchy location of the target class, assigning the first class name to the new class and assigning a second class name to the target class; and

(iii) adding instrumentation to the target class without modifying bytecode within the target class;

(c) causing a virtual machine to process as the target class the class assigned the first class name.

2. The method of claim 1 including the following by the steps of :

(aa) after completing step (a), operating a virtual machine to initiate loading and execution of the virtual machine executable software;

(ab) after completing step (aa) suspending the operation of the virtual machine after loading and before linking the at least one target class;

(ca) after completing step (c), un-suspending operation of the virtual machine.

3. The method of claim 1 including the following by the steps of :

(aa) after completing step (a), operating a virtual machine to initiate loading and execution of the virtual machine executable software;

(ab) after completing step (aa) interrupting the operation of the virtual machine after loading and before linking the at least one target class to execute a software program that executes step (c);

(ca) upon termination of the software program that executes step (c) resuming the operation of the virtual machine.

4. The method of claim 1 wherein the virtual-machine is a Java virtual machine and the step of identifying at least one target class included within the virtual machine executable software includes the steps of:

(a) specifying a set of class attribute names and associated value descriptions matching class attribute names and associated values possessed by at least one class included in the virtual-machine-executable software, the set including one or more of the following attribute names:

(i) a class name
(ii) an interface name
(iii) a parent class name
(iv) an inherited method name
(v) a defined method name
(vi) a private method name
(vii) an inherited field name
(viii) a defined field name
(ix) a private field name
(x) constant value attribute
(xi) synthetic attribute
(xii) code attribute
(xiii) exception attribute
(xiv) depreciated attribute

(b) searching for at least one class possessing class attribute names and associated values consistent with the specified class attribute names and associated value descriptions; and

(c) classifying the at least one class as a target class.

5. The method of claim 1 wherein the steps of suspending and un-suspending the operation of the virtual machine are performed using a interface provided by the virtual machine.

6. The method of claim 1 wherein the steps of suspending and un-suspending the operation of the virtual machine are performed using a virtual machine that is modified to perform this capability.

7. The method of claim 1 wherein the virtual machine is a Java virtual machine.

8. The method of claim 1 wherein the virtual-machine is a Java virtual machine and the steps of adding instrumentation to or modifying the at least one target class include adding, removing, modifying, reordering or substituting one or more of the following named class components:

(i) class name
(ii) super class name
(iii) interface index array
(iv) field table
(v) method table
(vi) constant pool
(vii) attribute table
(viii) index array
(ix) access flags

9. The method of claim 6 wherein the class component is represented by a plurality of memory locations, the step of modifying a class component including adding, removing, reordering or substituting one or more memory locations representing the class component or changing the contents of one or more memory locations representing the class component.

10. The method of claim 1 wherein the at least one target class is represented by a plurality of memory locations, the step of adding instrumentation to or modifying the at least one target class including adding, removing, or reordering or substituting one or more memory locations

representing the target class or changing the contents of one or more memory locations representing the at least one target class.

11. The method of claim 1 wherein the step of assigning the new class to a class hierarchy location adjacent to and above the class hierarchy location associated with the at least one target class includes:

modifying the new class to recognize a super class associated with the target class as the super class associated with the new class;

modifying the target class to recognize the new class as the super class associated with the target class.

12. The method of claim 1 wherein the step of assigning the new class to a class hierarchy location adjacent to and below the class hierarchy location associated with the at least one target class includes modifying the new class to recognize the target class, as the super class associated with the new class.

13. An apparatus for instrumenting virtual-machine-executable software, the software including one or more objects, each object being defined by a class, each class being associated with a class-hierarchy location, the method comprising the steps of:

(a) a target class identifier that identifies at least one target class included within the software, the at least one target class being associated with a first class-hierarchy location and with a first class name;

(b) an instrumenter that for each target class, adds instrumentation to the software according to at least one of the following steps:

(i) creating a new class, adding instrumentation to the new class, and assigning the new class to a class-hierarchy location adjacent to and above the first class-hierarchy location of the target class; and

(ii) creating a new class, adding instrumentation to the new class, and assigning the new class to a class-hierarchy location adjacent to and below the first class-hierarchy location of the target class, assigning the first class name to the new class and assigning a second class name to the target class; and

(iii) adding instrumentation to the target class without modifying bytecode within the target class;

(c) a virtual machine for processing as the target class the class assigned the first class name.

14. An apparatus for identifying at least one class included within virtual-machine-executable software, the apparatus comprising:

a class query engine which receives as input a set of class attribute names and associated value descriptions; and

a class searcher that sequentially searches classes included in software and performs a predetermined action upon identifying a class that possesses a set of class attribute names and associated values that match the set of class attribute names and associated value descriptions.